



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TR-603554

A Block-Based MC-SURE Algorithm for Denoising Sensor Data Streams

M. Ndoeye, C. Kamath

November 20, 2012

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

A Block-Based MC-SURE Algorithm for Denoising Sensor Data Streams

Mandoye Ndoye and Chandrika Kamath
 Center for Applied Scientific Computing
 Lawrence Livermore National Laboratory
 Livermore, California 94550
 Email: ndoye1,kamath2@llnl.gov

Abstract

We propose a strategy to automatically denoise sensor data streams corrupted with noise that can be approximated as additive white Gaussian noise. The proposed block-based method is adapted from the Monte-Carlo-SURE (MC-SURE) algorithm which enables the blind optimization of the denoising parameter of a wide class of filters. Our framework is formulated by identifying and addressing the challenges that arise when the MC-SURE algorithm is applied in an on-line data processing setting, where latency (and the length of data blocks) must be constrained. The strategy has been tested using real datasets. Our results indicate that the proposed method can be effectively used to handle the denoising of real sensor streaming data that reasonably fit the Gaussian model assumption.

I. INTRODUCTION

The increasing affordability of sensors is enabling their cost-effective use in real or near-real time monitoring of complex phenomena and systems. Some examples are fusion in Tokamak reactors [1], electric power grids [2] and large-scale infrastructure networks ([4], [3]). The measurements obtained from multiple sensors monitoring the phenomena or systems under consideration are analyzed to track their operating states or detect deviations from their normal behavior. The effectiveness of these real-time system monitoring algorithms depends, among other things, on the quality of their input data, that is, the level of the signal relative to the level of noise. Since sensor data are typically subject to non-negligible measurement errors from both instrumental and environmental noise sources, the quality of the measured data must be improved by filtering to remove most of the noise contributions while preserving important waveform information. Although a host of denoising algorithms already exist, most are not inherently designed for real (or near-real) time data processing as they often lack an on-line/automated mechanism for selecting the best parameter value for denoising an arbitrary measurement sequence. Some denoising algorithms, such as Kalman filters [5], are suited to on-line processing. Unfortunately, their performance depends on how well both the underlying processes that generated the observations and the noise statistics have been modeled. This requirement for a-priori full system model description is not practical in the general streaming data setting as it would be difficult to obtain an accurate model for each of the processes that generated the multiple data streams.

We develop a simple method for on-line denoising of a single sensor data stream with arbitrary waveform characteristics by extending the Monte-Carlo Stein's Unbiased Risk Estimate (MC-SURE) algorithm by Ramani *et al.* [6]. This is an approach that enables a blind optimization of the regularization parameters of a wide class of denoising algorithms that seek to recover an arbitrary signal corrupted by additive white Gaussian noise (AWGN). The proposed method is trivially applicable to denoising multiple data streams since the streams from distinct sensors are expected to have different signal and noise characteristics and can be processed independently. The MC-SURE formulation is particularly suited for use in denoising sensor data streams because it produces the optimal denoising parameters (using a mean-squared-error criterion) of a chosen filter without any assumptions about the underlying noise-free signal.

This technical report is organized as follows. In Section II, the general concept of the MC-SURE algorithm is described and relevant issues concerning its application to on-line/real-time data processing scenarios are discussed. Section III deals with the approaches taken to solve these problems. Experimental results and discussions are provided in IV, followed by a concluding summary in Section V.

II. CONCEPTS AND CHALLENGES

A. Description of the MC-SURE algorithm

The MC-SURE algorithm is a procedure for generating the optimal parameters of an arbitrary¹ denoising algorithm used to estimate a signal corrupted by additive white Gaussian noise. Since the optimality criterion is the minimum mean-square error (MSE), the MC-SURE strategy finds the best denoising parameter by minimizing Stein's Unbiased Risk Estimate [7], which is an unbiased estimate of the MSE that can be computed from measured data without any knowledge about the underlying

¹Any denoising function that is continuous and weakly differentiable. This coincides in practice with a wide class of denoising algorithms.

noise-free signal or the functional expression of the denoising filter. The formal setup is the following: consider the typical denoising problem where one has a noisy dataset

$$\mathbf{y} = \mathbf{s} + \mathbf{w}$$

comprised of $\mathbf{s} \in \mathbb{R}^N$, the vector of samples from a sought-after signal, and $\mathbf{w} \in \mathbb{R}^N$, the vector of the realizations of a zero-mean AWGN with variance σ^2 . The parametric denoising algorithm chosen is the mapping

$$\hat{\mathbf{s}}_\lambda = f_\lambda(\mathbf{y})$$

producing the signal estimate $\hat{\mathbf{s}}$ from the measured data \mathbf{y} . The vector-parameter, λ , controls the level of denoising and corresponds to the parameters of the denoising algorithm. The SURE statistic estimating the mean-squared error $\frac{1}{N} \|\mathbf{s} - \hat{\mathbf{s}}_\lambda\|^2$ is expressed in [6] as

$$T_\lambda(\mathbf{y}) = \frac{1}{N} \|\mathbf{y} - \hat{\mathbf{s}}_\lambda\|^2 + \frac{2\sigma^2}{N} \text{div}_{\mathbf{y}}\{f_\lambda(\mathbf{y})\} - \sigma^2 \quad (1)$$

where $\text{div}_{\mathbf{y}}\{f_\lambda(\mathbf{y})\}$ is the divergence of the mapping $f_\lambda(\cdot)$ with respect to the data \mathbf{y} . The optimal denoising parameter λ^* is the minimizer of the λ -dependent SURE curve $\{T_\lambda(\mathbf{y}) : \lambda \in \mathbb{R}^K\}$, where K is the number of elements in the vector-parameter λ . The optimal parameter is thus formally given by

$$\lambda^* = \arg \min_{\lambda} \{T_\lambda(\mathbf{y}) : \lambda \in \mathbb{R}^K\}, \quad (2)$$

and the best estimate of the underlying signal \mathbf{s} is given by $\hat{\mathbf{s}}^* = f_{\lambda^*}(\mathbf{y})$.

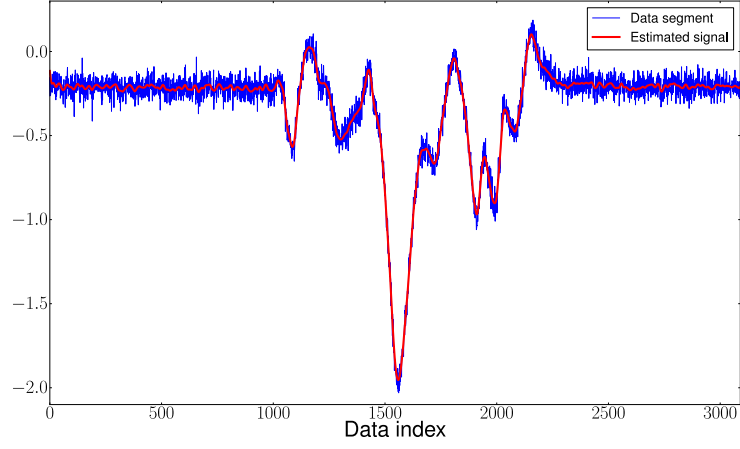
Figure 1 illustrates the effectiveness of the MC-SURE method for denoising a data segment corrupted with AWGN. Figure 1a shows a noisy data segment and the underlying signal estimated using MC-SURE. The example data segment is generated by adding AWGN to a known signal. The denoiser selected is the Gaussian filter and the denoising parameter λ is the standard deviation of the Gaussian filter. The MSE-statistic $\frac{1}{N} \|\mathbf{s} - \hat{\mathbf{s}}_\lambda\|^2$ is computed for 60 uniformly-spaced points on the range of λ values from 0 to 15. The MC-SURE method is used to compute the SURE-statistic $T_\lambda(\mathbf{y})$ for these same λ values. Figure 1b shows that the SURE curve approximates the MSE-curve and therefore can be used to find the λ parameter for use in denoising the example data segment, as shown in Figure 1a.

The MC-SURE algorithm extends the applicability of SURE-based denoising by introducing an effective approach for computing the divergence term $\text{div}_{\mathbf{y}}\{f_\lambda(\mathbf{y})\}$ without the need for a closed-form expression of the denoiser. This black-box approach only requires multiple responses of the denoising filter to synthesized inputs: the measured data \mathbf{y} plus multiple realizations of a zero-mean i.i.d. random vector with a suitably chosen variance ϵ . We chose the MC-SURE algorithm as the centerpiece of the proposed framework for denoising sensor data streams due to two key features. First, it provides a straightforward method for automatically optimizing the parameters of a denoising filter without any assumption about the desired underlying signal. Second, the algorithm works with a large set of denoising filters and thus provides flexibility when selecting denoising methods. However, the application of the MC-SURE method for on-line data processing has certain challenges, which must be solved in order to arrive at a working framework.

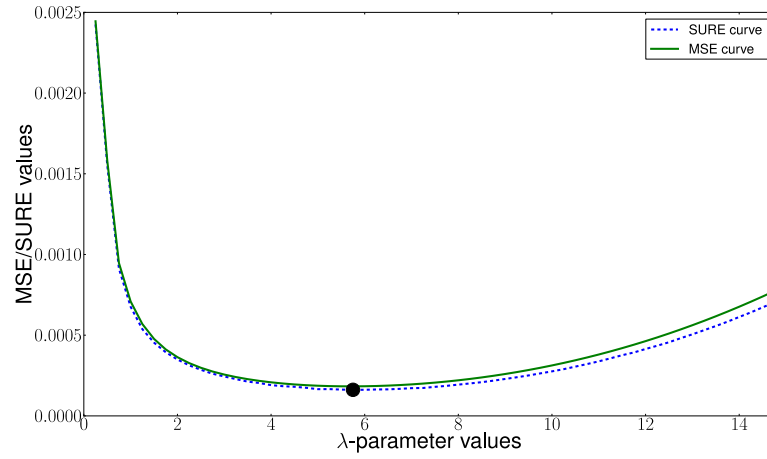
B. On-line implementation challenges

On-line data processing techniques are either sample-based or block-based [10]. In sample-based processing, the data are processed one sample at a time and the algorithm combines the information about its current state and the current sample to produce an output sample. For example, the recursive least squares (RLS) filter is an algorithm that does sample-based data processing. In block-based processing, the sequence of measurements is buffered into data blocks that are processed one at a time. Block-based processing distributes fixed computational overheads across multiple samples. However, this incurs an additional latency which would not be present in sample-based processing. Short Time Fourier Transform (STFT) computation is an example of block-based processing.

In this report, we consider the block-based approach for our on-line data processing framework. The critical need for minimizing latency dictates that the data blocks should have the minimally required number of observations, a requirement which introduces certain challenges in denoising streaming data using the MC-SURE algorithm. Our experimental explorations have also indicated other issues that are intrinsic to the MC-SURE method, rather than being particular to the on-line processing setting. We next discuss these challenges using the example dataset shown in Figure 2. This is a periodic dataset so that successive data blocks have the same frequency/information content, allowing more meaningful comparisons across multiple data blocks.



(a) Dataset and the estimated signal using the MC-SURE method.



(b) MSE and MC-SURE curves for the dataset. The dot is the minimizer of the SURE curve (and MSE-curve).

Fig. 1: Illustration of using the MC-SURE algorithm to denoise a dataset corrupted by AWGN using a Gaussian filter.

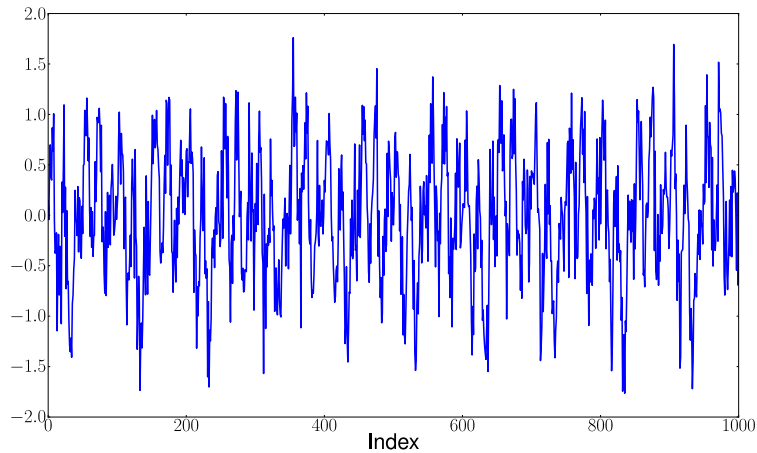


Fig. 2: Segment of the example dataset used to illustrate issues/challenges. The dataset is synthesized by adding AWGN (with standard deviation $\sigma = 0.34$) to a sum of sinusoidal functions: $f(x) = \frac{1}{3} \cdot \cos(20x) + \frac{1}{3} \cdot \cos(10x - \pi/4) - \frac{1}{3} \cdot \sin(5x) + \frac{1}{2} \cdot \sin(30x - \pi/4)$.

1) **Detrimental effects of noise estimation errors:** The SURE statistic in Equation 1 considers the noise standard deviation σ to be exactly known. This is clearly not the case in practical settings where this parameter must be estimated. In [6], Ramani *et al.* assumed that the parameter σ can be reliably obtained using the Donoho median estimator [8], which estimates the noise standard deviation of a length- N dataset \mathbf{y} as

$$\hat{\sigma} = \frac{M_{\mathbf{y}}}{0.6745},$$

where $M_{\mathbf{y}}$ is the median of the $N/2$ wavelet coefficients at the finest scale. The test datasets used in [6] consisted of images corrupted with synthesized AWGN. In such cases, the noise is perfectly white/Gaussian and the number N of observations is sufficiently large (typically $N \geq 256^2$) to assume that the estimate of σ is practically equal to its true value.

However, in the context of real datasets, the observed noise is typically not AWGN, although it can be reasonably modeled as such in some instances. In streaming data processing, the limited number of samples N from a given data block \mathbf{y} might be too small to reliably estimate the associated noise parameter. Unfortunately, uncertainties about the noise parameter can lead to unreliable estimation of the SURE curves as shown in the following experiment.

From the example data of Figure 2, we generate two groups of datasets. The first is comprised of 20 consecutive length-25 data segments $\{\mathbf{y}_i^{(1)} : i = 1, \dots, 20\}$ while the second, of 20 consecutive length-300 datasets $\{\mathbf{y}_i^{(2)} : i = 1, \dots, 20\}$. Individual segments from a group have practically the same frequency content and noise characteristics since the example data they are taken from is periodic. Thus, the MC-SURE algorithm should ideally report the same denoising parameter value for all 20 data segments. For each length-25 data segment $\mathbf{y}_i^{(1)}$, we generate an estimate $\sigma_i^{(1)}$ of the noise parameter σ using the Donoho median estimator [8]. This estimate is used in Equation 1 to compute the SURE curve, resulting in the sequence of SURE-statistics $T_{\lambda}(\mathbf{y}_i^{(1)})$ for λ ranging from 0.0 to 15.0 at 0.25-increment. The parameter λ is the standard deviation of the Gaussian filter which is chosen here as our example denoiser. The marked/blue curves in Figure 3 correspond to the SURE curves for the 20 length-25 data segments. We observe that the computed SURE curves vary significantly from one another although they intend to estimate the same MSE-curve. Further, SURE curves from two different data blocks can report drastically different values for the best denoising parameter to select. For example in Figure 3, the minimizer is equal to 0.25 for one curve but 2.00 for another. In contrast, when we consider the SURE curves of the 20 length-300 data segments $\{\mathbf{y}_i^{(2)} : i = 1, \dots, 20\}$ (the solid/red curves in Figure 3), we find better agreement and the minimizers are now all in the vicinity of $\lambda = 2.0$.

Unfortunately, we cannot arbitrarily increase the size of data blocks in on-line data processing settings due to the need for low latency. Therefore, to make the MC-SURE approach work for streaming data, a different strategy is needed to generate sufficiently reliable SURE curves.

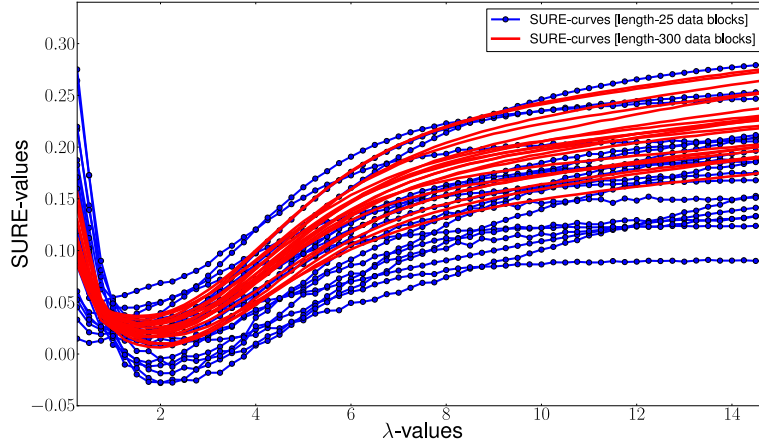


Fig. 3: SURE curves for two groups of data segments: 20 consecutive length-25 data segments and 20 consecutive length-300 data segments

2) **Reliable computation of the divergence term:** Given a length- N data segment \mathbf{y} and a specific (vector)-parameter value λ_j from a suitably chosen grid of denoising parameters $\{\lambda_j : j = 1, 2, \dots, J\}$, the reliability of the computed SURE-statistics $T_{\lambda_j}(\mathbf{y})$ also depends on the accuracy of the divergence term $\text{div}_{\mathbf{y}}\{f_{\lambda_j}(\mathbf{y})\}$. It has been shown in [6] that the divergence can be expressed as

$$\text{div}_{\mathbf{y}}\{f_{\lambda_j}(\mathbf{y})\} = \lim_{\epsilon \rightarrow 0} E_{\mathbf{b}}\{\mathbf{b}^T(f_{\lambda_j}(\mathbf{y} + \mathbf{b}) - f_{\lambda_j}(\mathbf{y}))\}, \quad (3)$$

where \mathbf{b} is a zero-mean i.i.d. random vector with covariance $\epsilon^2 \mathbf{I}$, and $E_{\mathbf{b}}\{\cdot\}$ denotes expectation with respect to \mathbf{b} . When a sufficiently small value of ϵ is chosen, $\text{div}_{\mathbf{y}}\{f_{\lambda_j}(\mathbf{y})\}$ is estimated using a Monte-Carlo (MC) approach, which first generates

k realizations $\{\mathbf{b}_i\}_{i=1}^k$ of the length- N random vector \mathbf{b} and then computes the k -MC-run divergence estimate as

$$\widehat{\text{div}}_{\mathbf{y}}^{(k)}\{f_{\lambda_j}(\mathbf{y})\} = \frac{1}{\epsilon k} \sum_{i=1}^k \mathbf{b}_i^T (f_{\lambda_j}(\mathbf{y} + \mathbf{b}_i) - f_{\lambda_j}(\mathbf{y})), \quad (4)$$

which amounts to averaging the k single-MC-run divergence estimates.

To generate a SURE curve for a data segment \mathbf{y} , the divergence is estimated for each element of the parameter grid $\{\lambda_j : j = 1, 2, \dots, J\}$. We conducted the following experiments to understand how the number of runs, k , and the length of the data segment, N , influence the estimate of the divergence. We first define the estimation error as

$$E_{\lambda_j}^{(k)}(\mathbf{y}) = \widehat{\text{div}}_{\mathbf{y}}^{(k)}\{f_{\lambda_j}(\mathbf{y})\} - \text{div}_{\mathbf{y}}\{f_{\lambda_j}(\mathbf{y})\} \quad (5)$$

for $i = 1, 2, \dots, J$, where the reference divergence-curve $\text{div}_{\mathbf{y}}\{f_{\lambda_j}(\mathbf{y})\}$ is the sequence to which the k -MC-run divergence estimate converges to in probability. For the example dataset, the reference divergence is obtained by using a very large number (i.e., $k = 50000$) of MC simulations. As k increases, the variances of estimation errors tend to zero.

Figure 4 illustrates how the error variance changes with N when we estimate the divergence curve with a single MC run. For each parameter λ_j , the associated error variance is estimated using 1000 computed values of $E_{\lambda_j}^{(1)}(\mathbf{y})$. The lowermost curve of Figure 4, which corresponds to the variances of $\{E_{\lambda_j}^{(1)}(\mathbf{y}) : j = 1, 2, \dots, J\}$ for a data segment of length $N = 9600$, indicates that the error-variances are relatively small across the entire range of λ_j values. The remaining curves correspond to the error variance for $N = 4800, 2400, 1200, 600$, and 300 . We observe that as N decreases, the error variances increase rapidly and can become significant. Further, the error variance increases as λ_j decreases to zero indicating that the number of MC runs required to attain a specific approximation error may depend on the value of λ_j .

Figure 5 shows 50 single-MC-run estimates of the divergence curve for the case when $N = 300$. The individual estimates deviate significantly from the reference curve, indicating that multiple single-MC-run estimates must be combined to produce a reasonable approximation of the divergence curve.

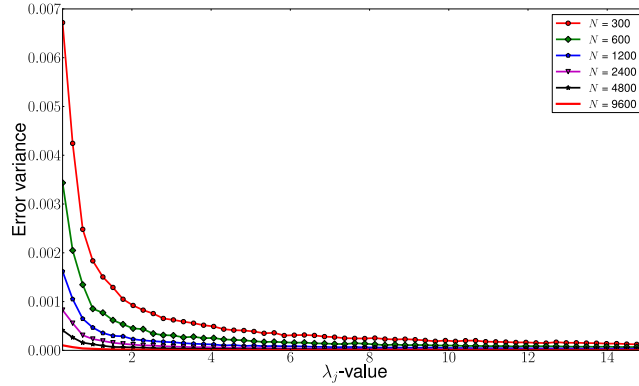


Fig. 4: Variance of the error in estimating the divergence curve for different value of the data size N .

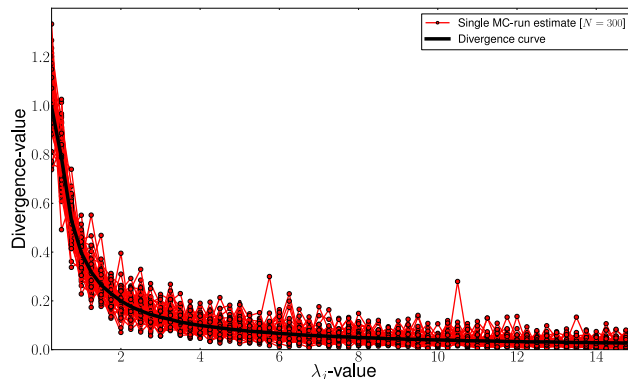


Fig. 5: Comparison of the divergence curve with single-MC-run estimate when the data size is relatively low: $N = 300$ samples.

Therefore, to adapt the MC-SURE algorithm to streaming data, where we process blocks of small size, we need to automatically choose a number k of MC runs that can 1) be averaged within the available time frame and 2) lead to a reliable divergence-curve estimate which can be used to compute a sufficiently accurate SURE curve and generate an appropriate denoising parameter.

3) **Effects of strong DC components::** Our exploratory experiments suggested that using the MC-SURE procedure on data blocks with a strong DC components failed to return an adequate denoising parameter value. To illustrate this issue, consider the two datasets \mathbf{z} and \mathbf{z}_D , where \mathbf{z} is a length-300 segment of the example data and \mathbf{z}_D is generated by adding the constant-vector \mathbf{D} to \mathbf{z} : $\mathbf{z}_D = \mathbf{D} + \mathbf{z}$, where all elements of \mathbf{D} are set equal to $D = 10$. The application of the MC-SURE algorithm to \mathbf{z} produces the SURE curve in Figure 6(a), which reports a filter parameter value of $\lambda^* = 2$. Since \mathbf{z}_D and \mathbf{z} are identical except for the presence of a DC term, we expect the optimal parameter values for denoising to be similar. However when MC-SURE is applied to \mathbf{z}_D , the resulting SURE curve, shown in Figure 6(b), indicates that no denoising is needed as $\lambda^* = 0$. The presence of the DC component erroneously decreases the parameter value reported by the MC-SURE algorithm. In fact, we found that if the DC content is very high, the MC-SURE algorithm appears to always report that no denoising is needed regardless of the amount of noise present. Since many interesting real datasets (e.g., wind power generation time series) have been found to contain strong DC components, this issue must be addressed in practice. Contrary to the two issues mentioned previously, this problem does not result from on-line data processing settings.

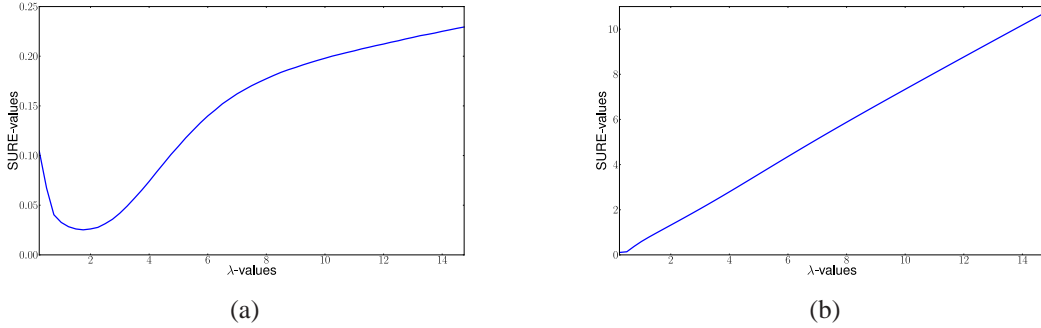


Fig. 6: Comparison of the SURE curves generated using the datasets \mathbf{z} and \mathbf{z}_D . (a) SURE curve of \mathbf{z} and (b) SURE curve of \mathbf{z}_D plus DC-component. Although the two SURE curves differ significantly, \mathbf{z} and \mathbf{z}_D only differ by a DC-component term.

4) **Spatial resolution issues:** We previously discussed the issues that arise in computing the noise parameter σ and the divergence-term $\text{div}_{\mathbf{y}}\{f_{\lambda}(\mathbf{y})\}$ in Equation 1. We next evaluate the issue of reliably computing the remaining term $\frac{1}{N}\|\mathbf{y} - \hat{\mathbf{s}}_{\lambda}\|^2$. Figure 7 shows the SURE curves for data blocks of different lengths extracted from the same sum of sinusoids function plus AWGN noise with variance σ^2 . The individual SURE curves are generated using the same value of σ and a common reference divergence-curve. Since Gaussian denoising is a linear operation, the divergence-curve is the same for all data sequences considered [6]. The discrepancies observed between the SURE curves can only be attributed to the term $\frac{1}{N}\|\mathbf{y} - \hat{\mathbf{s}}_{\lambda}\|^2$. For small data blocks ($N = 10$), the SURE curve reports that no denoising is needed ($\lambda^* = 0$). As N increases, the SURE curves are more in agreement and report similar denoising parameter values. This experiment indicates that even if the noise and divergence terms are well approximated, we need a minimal number of observations in each data block so that the algorithm can distinguish the signal from the higher frequency noise in the data.

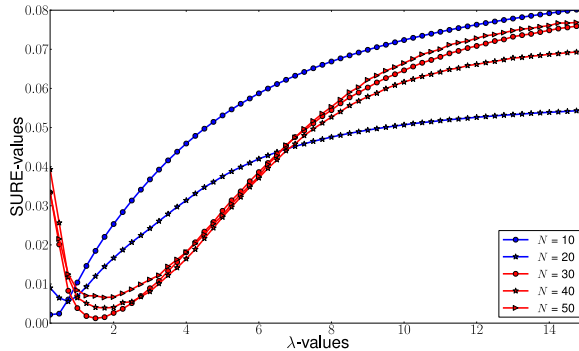


Fig. 7: Illustration of the potential effects of spatial resolution on the accurate computation of SURE curves.

III. A FRAMEWORK FOR STREAMING DATA: METHODOLOGY AND IMPLEMENTATIONS

We next describe a framework for denoising streaming data that addresses the issue we identified in Section II. Consider a data stream $\mathbf{y} = \{y_n : n = 0, 1, \dots\}$ comprised of a desired signal $\mathbf{s} = \{s_n : n = 0, 1, \dots\}$ plus additive white Gaussian noise. The data model assumes that the noise standard deviation may vary slowly with time. The streaming measurements, \mathbf{y} , are taken in as successive overlapping data blocks of judiciously chosen sizes. The methodology for selecting the data block sizes will be detailed later. Consecutive blocks are overlapped to mitigate possible edge effects which could result from the filtering of limited-size data blocks. The i -th data block is denoted by

$$\mathbf{y}_i = \{y_n : N_i \leq n < M_i\}$$

with $N_{i+1} = M_i - L$, where L is the length of the fixed overlap between two consecutive data blocks.

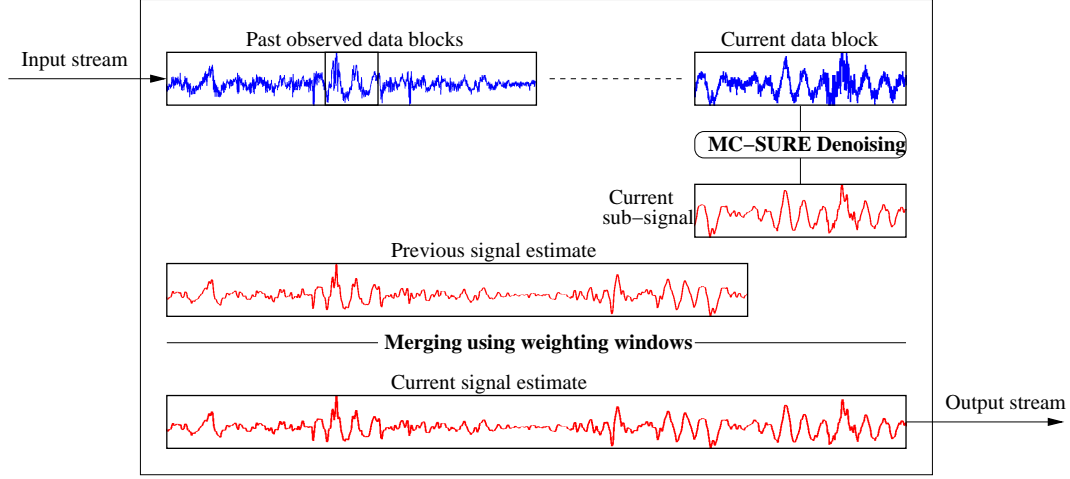


Fig. 8: Schematic representation of the block-based, on-line, data processing framework centered on the MC-SURE algorithm.

Figure 8 shows a schematic representation of the data processing framework where the sequence of data blocks $\{\mathbf{y}_i : i = 0, 1, \dots\}$ is processed one block at a time. The noise in the i -th data block \mathbf{y}_i is removed using the selected denoising filter $f_\lambda(\cdot)$ with λ set to λ_i^* , the best MC-SURE-based parameter value for denoising \mathbf{y}_i . The processing of the i -th input data block \mathbf{y}_i returns the i -th output data block

$$\hat{\mathbf{s}}_i = f_{\lambda_i^*}(\mathbf{y}_i),$$

which is the sub-signal of the desired signal \mathbf{s} that is contained in \mathbf{y}_i . Thus, the processing of the sequence of data blocks $\{\mathbf{y}_i : i = 0, 1, \dots\}$, results in a sequence of sub-signals estimates $\{\hat{\mathbf{s}}_i : i = 0, 1, \dots\}$.

We denote by $\hat{\mathbf{s}}^{(\mathcal{I}+1)}$ the estimate of the entire portion of the signal \mathbf{s} that is available after the processing of the $(\mathcal{I}+1)$ -th data block $\mathbf{y}_{\mathcal{I}+1}$. The signal estimate $\hat{\mathbf{s}}^{(\mathcal{I}+1)}$ can be formed by properly stitching together the existing items in the collection of overlapping sub-signal estimates $\hat{\mathbf{s}}_0, \hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2, \dots, \hat{\mathbf{s}}_{\mathcal{I}+1}$ as follows:

- $\hat{\mathbf{s}}^{(\mathcal{I})} = \{\hat{s}_n : n = 0, 1, 2, \dots, M_{\mathcal{I}}\}$ is the previously available signal estimate (i.e., from stitching together the first $\mathcal{I}+1$ sub-signal estimates).
- $\hat{\mathbf{s}}_{\mathcal{I}+1} = \{\hat{s}_n : n = M_{\mathcal{I}} - L, \dots, M_{\mathcal{I}+1}\}$ is the new sub-signal estimate, generated by denoising the $(\mathcal{I}+1)$ -th data block.
- The updated signal estimate $\mathbf{s}^{(\mathcal{I}+1)}$ is constructed by using tapered windows to properly merge the new sub-signal estimate $\hat{\mathbf{s}}_{\mathcal{I}+1}$ to the previous estimate $\mathbf{s}^{(\mathcal{I})}$ via

$$\mathbf{s}^{(\mathcal{I}+1)} = \mathbf{s}^{(\mathcal{I})} \odot \mathbf{W}^{(\mathcal{I})} + \hat{\mathbf{s}}_{\mathcal{I}+1} \odot \mathbf{W}_{\mathcal{I}+1}.$$

This procedure is initialized by setting $\mathbf{s}^{(0)} = \hat{\mathbf{s}}_0$. The operator \odot denotes element-by-element multiplication. The terms $\mathbf{W}^{(\mathcal{I})}$ and $\mathbf{W}_{\mathcal{I}+1}$ are one-sided tapering windows given by

$$\mathbf{W}^{(\mathcal{I})} = \begin{cases} 1, & 0 \leq n < M_{\mathcal{I}} - L \\ \beta_2(n - (M_{\mathcal{I}} - L)), & M_{\mathcal{I}} - L \leq n < M_{\mathcal{I}} \end{cases}$$

$$\mathbf{W}_{\mathcal{I}+1} = \begin{cases} \beta_1(n - (M_{\mathcal{I}} - L)), & M_{\mathcal{I}} - L \leq n < M_{\mathcal{I}} \\ 1, & M_{\mathcal{I}} \leq n < M_{\mathcal{I}+1} \end{cases}$$

where $\beta_1(n)$ and $\beta_2(n)$ are respectively the first- and second-half of a length $2L$ cosine tapering window, though other types of tapering windows could be also used. The use of tapering windows allows us to mitigate edge effects.

The performance of this system depends on how effectively individual data blocks are denoised. We next describe the steps we have taken to address the issues that arise when applying the MC-SURE algorithm to short data segments. Additionally, we describe the methodology for selecting the length of data blocks.

A. Removal of DC components

We have previously shown that the performance of the MC-SURE algorithm will degrade if the data block \mathbf{y}_i has a non-negligible DC-component. To mitigate the issue and obtain a good parameter value for denoising the data, we filter \mathbf{y}_i with a notch-filter centered at the DC-frequency [9]. If the data block \mathbf{y}_i has no DC-component, it will be indifferent to this DC-blocking filter. We may assume in the following discussions that the i -th data block \mathbf{y}_i is free of a DC-component since, if it had one, the DC-blocking filter would have removed it at this stage.

B. Noise estimation

The reliability of the SURE curve for the i -th data block \mathbf{y}_i depends, among other things, on how accurately the noise parameter σ is estimated. Unfortunately, a sole reliance on the limited number of measurements from \mathbf{y}_i is expected to produce an insufficiently accurate estimate of σ . We mitigate the problem by combining information from multiple data blocks. Since the data model expects the noise characteristics to change slowly with time, it is reasonable to assume that M consecutive data blocks have additive white Gaussian noise with the same standard deviation σ . Using the Donoho median estimator, a separate estimate of σ is computed for each of the M data blocks and the estimate $\bar{\sigma}_{\mathcal{I}}$ for the current data block $\mathbf{y}_{\mathcal{I}}$ is obtained via weighted averaging

$$\bar{\sigma}_{\mathcal{I}} = \frac{\sum_{i=\mathcal{I}-M+1}^{\mathcal{I}} N_i \hat{\sigma}_i}{\sum_{i=\mathcal{I}-M+1}^{\mathcal{I}} N_i}$$

where $\{\hat{\sigma}_{\mathcal{I}-M+1}, \hat{\sigma}_{\mathcal{I}-M+2}, \dots, \hat{\sigma}_{\mathcal{I}}\}$ are the noise estimates from the previous M data blocks, N_i denotes the length of the i -th data block and estimates from longer data blocks have larger weights as they are expected to be more reliable.

C. Computing the divergence curve

The estimate of the divergence for the current data block $\mathbf{y}_{\mathcal{I}}$ using k MC runs is given by

$$d_j^{(k)} = \widehat{\text{div}}_{\mathbf{y}_{\mathcal{I}}}^{(k)} \{f_{\lambda_j}(\mathbf{y}_{\mathcal{I}})\}, \quad j = 1, 2, \dots, J$$

and computed using Equation 4. The parameter λ_j is the j -th element in the chosen grid of denoising parameter values. For a fixed j , $\{d_j^{(k)} : k = 1, 2, \dots\}$ forms a convergent sequence of the estimates of the j -th value of the divergence-curve, which is denoted here by d_j . Since the error $|d_j^{(k)} - d_j|$ tends to zero as k increases, the j -th divergence value could then be estimated by $d_j^{(K_j)}$ with K_j chosen such that

$$r_j^{(K_j)} = \frac{|d_j^{(K_j+1)} - d_j^{(K_j)}|}{|d_j^{(2)} - d_j^{(1)}|} < \delta, \quad (6)$$

where δ is the convergence stopping criterion. However, individually identifying the number of MC runs for each of the J divergence-values could lead to practical difficulties as the convergence rate of $\{r_j^{(k)} : k = 1, 2, \dots\}$ is not guaranteed to be the same for different j . Thus, instances might occur where, for one or a small number of j -indexes, an unreasonably large number of runs is needed to satisfy Equation 6. To circumvent this situation, we seek an average/group convergence requirement, where for all parameter values λ_j , the divergence-value is estimated by $d_j^{(K)}$ with K such that

$$r^{(K)} = \frac{1}{J} \sum_{j=1}^J r_j^{(K)} < \delta. \quad (7)$$

The k -indexed sequence $r^{(k)}$ starts with the value of one, and decreases toward zero as the divergence-curve estimate $\{d_j^{(k)} : j = 1, 2, \dots, J\}$ converges with additional MC runs. Figure 9 shows an example profile for the sequence $r^{(k)}$, where the choice of $\delta = 0.02$ leads to the termination of the divergence curve estimation after $K = 67$ MC simulations.

D. Calculating the denoised signal

Once we have obtained the sequence $\{d_j^{(K)} : j = 1, 2, \dots, J\}$ of estimated divergence values and the estimated noise parameter value $\hat{\sigma}$, we can generate the sequence of SURE-values $\{T_{\lambda_j}(\mathbf{y}_{\mathcal{I}}) : j = 1, 2, \dots, J\}$ using Equation 1. The best parameter for denoising the current data block is given by

$$\lambda_{\mathcal{I}}^* = \arg \min_{\lambda_j} \{T_{\lambda_j}(\mathbf{y}_{\mathcal{I}}) : j = 1, 2, \dots, J\}$$

and the associated sub-signal estimate is given by

$$\mathbf{s}_{\mathcal{I}} = f_{\lambda_{\mathcal{I}}^*}(\mathbf{y}_{\mathcal{I}})$$

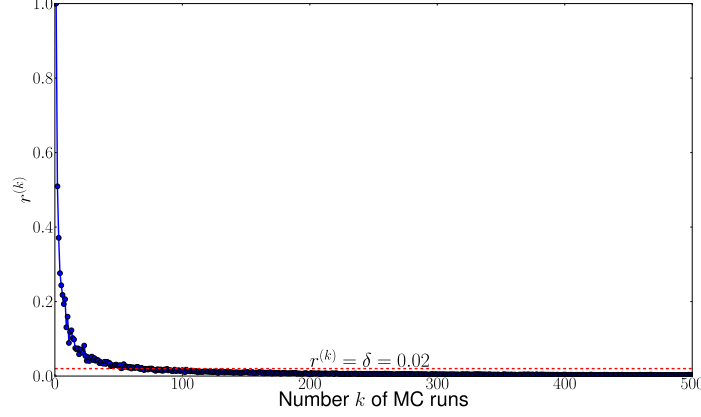


Fig. 9: Illustration of the profile of the sequence $r^{(k)}$, which is used to track the convergence of the divergence curve estimate.

E. Choosing the size of the next data block

To guarantee that spatial aliasing is minimized (i.e., frequency components in a data block are easily distinguishable), the size of each data block (except the first one) is selected using information from previously observed data. At the \mathcal{I} -th data block, we use principles of scale-space theory [11], [12] to estimate the typical size of the dominant signal feature in the block. By choosing the size of the $(\mathcal{I}+1)$ -th block significantly larger than this feature size, we ensure that we have enough samples to reasonably discriminate the frequency components present in the data. Since data processing latency has to remain minimal, the size of data blocks must be bounded above. In addition, the sizes should also be bounded below to have sufficient observations to estimate the noise parameters accurately. At the \mathcal{I} -th block, the size of the next data block is thus determined via

$$N_{\mathcal{I}+1} = \min \{N_{\max}, \max \{\theta \cdot \Delta_{\mathcal{I}}, N_{\min}\}\} \quad (8)$$

where

- $\Delta_{\mathcal{I}}$ is the data scale at the current data block that is computed via scale-space theory techniques and gives information on the current expected size of the most relevant features in the data.
- θ is a multiplicative factor that gives the data block size from computed data scale.
- N_{\min} is the minimally allowable data block size
- N_{\max} is the maximally allowable data block size.

IV. RESULTS AND DISCUSSIONS

We next discuss results from an experiment where the proposed framework is tested using a real dataset that is comprised of sampled measurements of the body acceleration response of a vehicle traveling on a road (see Figure 10). These measurements were taken at the sampling rate of 200 data points per second. Table I lists the parameters that are to be set by the user when implementing the proposed strategy, along with the specific values chosen in the denoising of our specific dataset.

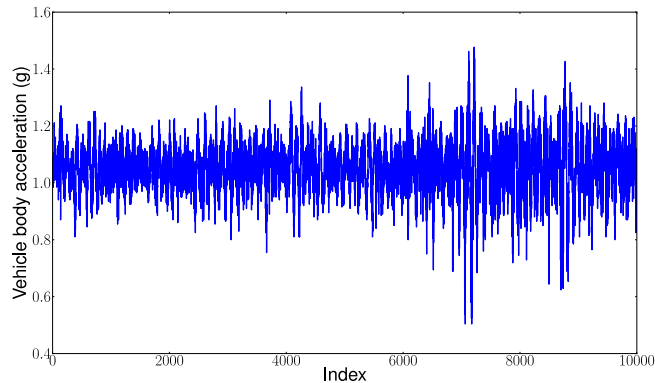


Fig. 10: Example real dataset: vehicle body acceleration measurements taken at the rate of 200 samples per second.

TABLE I: Parameter values used in the implementation of the proposed strategy for the example real dataset.

Parameter	Numerical Value
Standard deviation of probing noise: ϵ	0.0002
Maximally allowable number of MC simulations : N	400
Stopping criterion for the MC-SURE curve convergence: δ	0.02
Maximally allowable segment size: N_{\max}	500
Minimally allowable segment size: N_{\min}	30
Fixed overlap between consecutive segments: L	15
Multiplicative factor: θ	12
Noise estimation parameter M	5

We next introduce approaches for selecting the values of parameters for a dataset.

- Jointly selecting N and δ via calibration:** The appropriate number of MC runs, N , is system dependent and the appropriate stopping criterion δ depends on the dataset under consideration. As N increases, the accuracy of the SURE curve is guaranteed to increase, though it would take longer to generate the result. This is similarly true when the parameter δ decreases. There is an inter-dependence between the choices of N and δ - the computation of the SURE curve will terminate if either the stopping criterion δ is reached or all N simulations are done. Hence, they are selected jointly. Given a class of datasets, the user needs to use a calibration process to find the values of N and δ that give sufficiently repeatable results since it is impossible to a-priori select the appropriate values for these variables for an arbitrary dataset. This calibration can be done by using a sample measurement segment taken from historical data. In Figure 11(a), repeated runs of the denoising procedure on the same dataset lead to unsatisfactory and non-repeatable results as a portion of the signal (i.e., around index-700 to index-900) is not properly denoised for some runs. In Figure 11(b) where $N = 400$ (i.e., N is larger) and $\delta = 0.02$ (i.e., δ is smaller), we observe that the many runs are reasonably similar and all are properly denoised. These choices of parameters would thus used in the implementation.

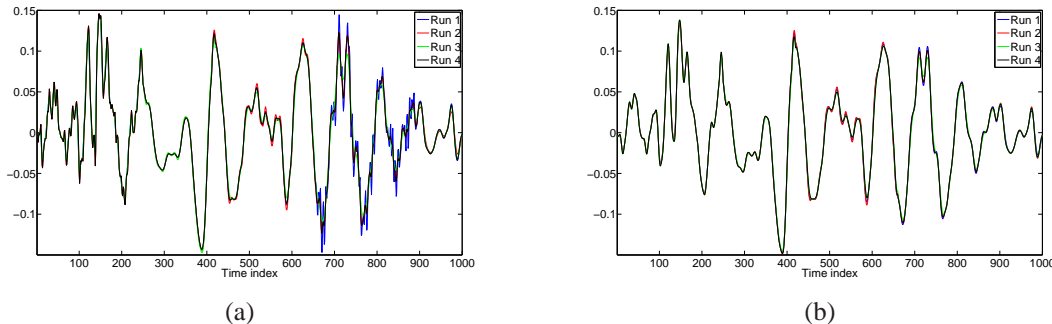


Fig. 11: Repeatability of denoising results as a function of the choices of the parameters N and δ . (a) Results when $N = 100$ and $\delta = 0.5$. (b) Results when $N = 400$ and $\delta = 0.02$.

- Selection of ϵ :** The parameter ϵ is the standard deviation of the synthesized probe noise used in the MC simulations. We confirmed the observation of Ramani *et al.* [6] that the MC-SURE denoising procedure is robust to even very large changes in the value of ϵ . Figure 12(a) shows a noisy data segment and Figure 12(b) shows its denoised versions with ϵ equal to 1.0 and 10^{-12} . The results are very similar though ϵ has changed by twelve orders of magnitude. Thus, for all practical purposes, the user should choose any reasonable value of ϵ in the above range. However, since ϵ is involved in a limit computation, it is preferable in principle for it to be as small as possible. But, selecting ϵ too small could lead to numerical round-off errors. To guard against these potential issues, we recommend choosing in general ϵ to be in the vicinity of 10^{-6} , which is the midpoint of the interval $[1.0, 10^{-12}]$ in the logarithmic scale.
- Choosing N_{\max} :** The size of the first data block must be preset by the user since there are no preceding blocks available to automatically compute the size. We set this size to twice N_{\max} , which is the maximally allowable block size for the dataset under consideration. The value of N_{\max} must be derived from a-priori expected data scale information. Since scale space theory uses the Gaussian kernel, the expected sizes of the most salient features in the data would be in the range of $6 \times \sigma_d$, where σ_d is the expected data scale. The parameter σ_d can be computed from a large enough sample of historical data (if available) or given from user experience. To guarantee that N_{\max} is sufficiently large and have good denoising performance, it has been found that choosing N_{\max} equal to $2 \times 6 \times \sigma_d$ (or larger) works well as illustrated via Figure 13. The expected scale σ_d of the vehicular response data is equal to 15 and thus N_{\max} should be set to 180 or larger. We observe that when N_{\max} is less than 180, the denoising performance is unsatisfactory, whereas good denoising performance is obtained for N_{\max} greater than 180.

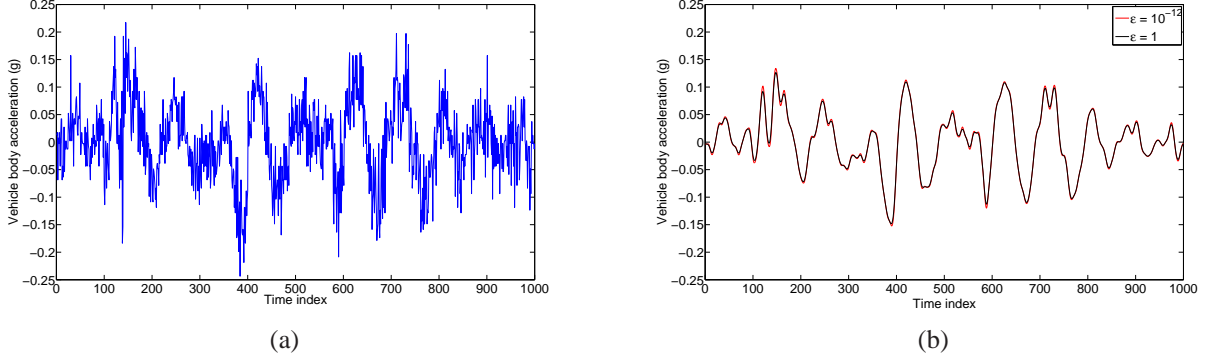


Fig. 12: Illustrating the robustness of the MC-SURE algorithm to large changes in the value of the noise probe parameter ϵ . (a) Raw dataset. (b) Denoising results with ϵ is equal to 10^{-12} and 1.0.

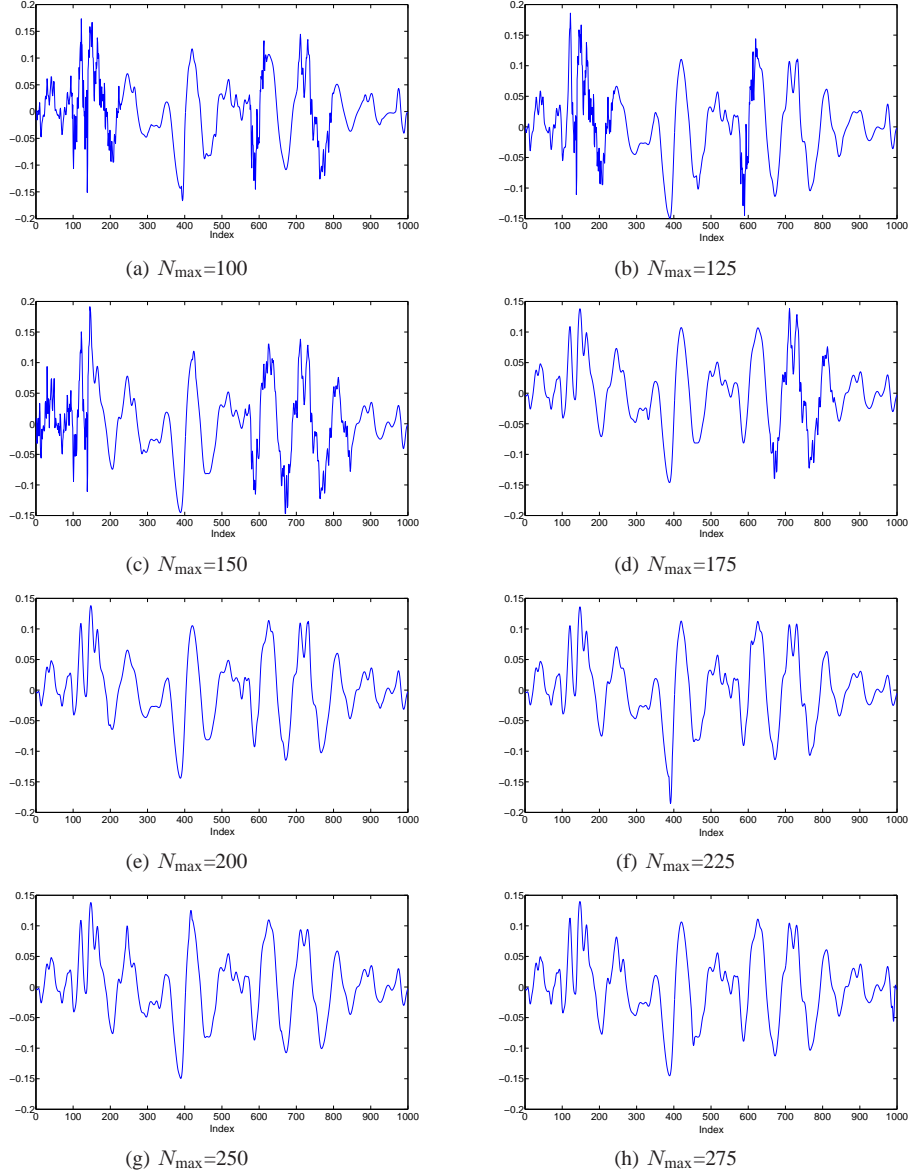


Fig. 13: An illustration of the choice of the parameter N_{\max} on denoising performance of the algorithm.

- **Selecting N_{\min} and L :** The choice of the minimally allowable data block size N_{\min} could affect the accuracy of the noise power estimation. So, if any computed data block length is less than N_{\min} , it is set to N_{\min} . If N_{\min} is too small, the noise power in a data block cannot be accurately estimated and we would need to consider more data blocks (that is, larger M) to improve the estimation. There is thus an inter-dependence between the choices of N_{\min} and M . Since the expected data scale is obtained when choosing N_{\max} , we also set N_{\min} to be a function of data scale, $N_{\min} = 2 \times \sigma_d = N_{\max}/6$. The overlap parameter is then chosen equal to $L = N_{\min}/2$. To assess the reliability/robustness of these choices using the available data, we change the values of N_{\min} to assess sensitivity. We observe in Figure 14 that with an adequate choice of parameters, the proposed approach for setting N_{\min} leads to good performance results that are not very sensitive to changes in the selected values.

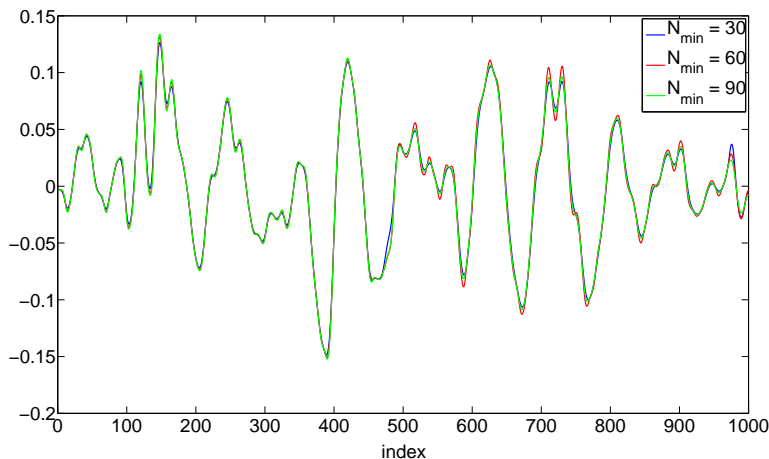
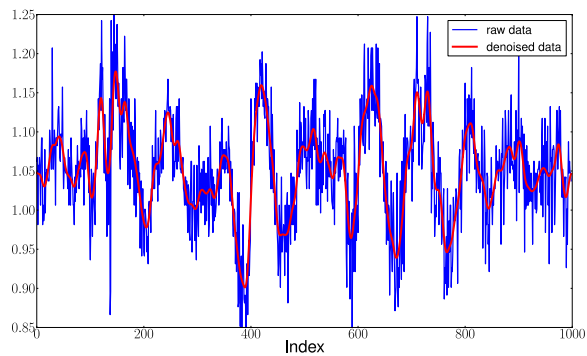


Fig. 14: Assessment of the sensitivity of the choice of N_{\min} using the available data.

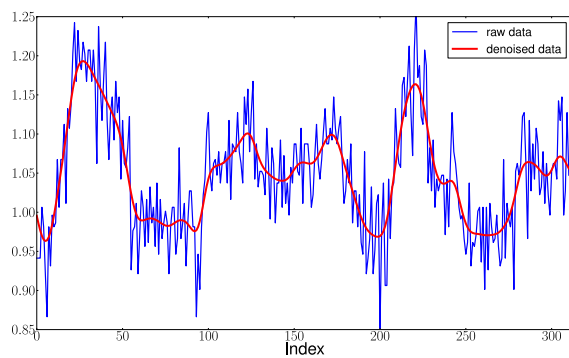
Finally, we present the results for denoising our example dataset using a Gaussian filter as the example denoiser. Table II shows a listing of the computed statistics and parameters for 20 consecutive data blocks. The parameter λ_i^* in the table is the parameter of the Gaussian filter as determined by the MC-SURE algorithm. Characteristic results of the proposed automated block-by-block data denoising procedure are shown in Figure 15 using sample data blocks. The results seem to indicate that good denoising performance is attainable from real datasets with noise that reasonably approximate the AWGN model, which is assumed by the MC-SURE approach.

TABLE II: Listing of computed statistics/parameters for 20 consecutive overlapping data segments.

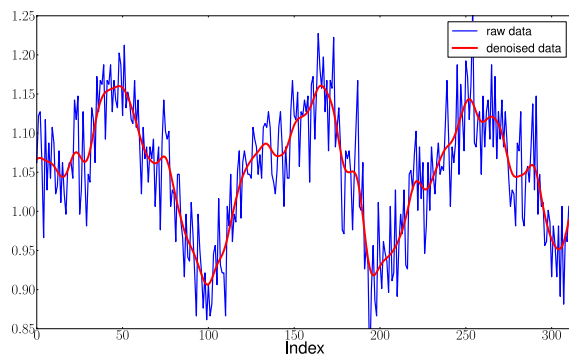
Block	Scale	Length	$\hat{\sigma}_i$	$\bar{\sigma}_i$	λ_i^*
1	12	1000	0.0396	0.0396	3.75
2	12	244	0.0553	0.0427	4.25
3	15	244	0.0480	0.0435	7.0
4	19	360	0.0357	0.0420	5.0
5	13	456	0.0398	0.0416	3.25
6	13	456	0.0464	0.0440	4.25
7	11	312	0.0405	0.0419	3.25
8	15	312	0.0527	0.0429	3.25
9	12	360	0.0476	0.0451	3.75
10	13	360	0.0381	0.0451	3.25
11	31	312	0.0323	0.0423	3.75
12	13	500	0.0368	0.0411	2.75
13	14	500	0.0322	0.0372	3.0
14	13	336	0.0382	0.0355	3.5
15	13	336	0.0393	0.0357	3.5
16	15	312	0.0381	0.0366	1.0
17	13	360	0.0486	0.0388	0.75
18	11	360	0.0590	0.0450	0.5
19	12	312	0.0554	0.0483	3.75
20	14	288	0.0650	0.0533	0.5



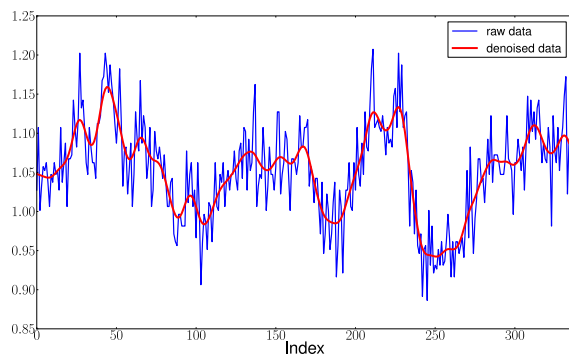
(a) 1-st data block.



(b) 7-th data block.



(c) 8-th data block.



(d) 14-th data block.

Fig. 15: Denoising results in some sample data blocks.

V. SUMMARY

In this report, we extended the MC-SURE algorithm to automatically denoise streaming data that have been corrupted with Gaussian or approximately-Gaussian noise. To obtain a practical solution approach, we first recognized that the MC-SURE algorithm is well-suited for streaming data as it prescribes a blind procedure for optimizing the regularization parameters of most of the commonly-used denoising filters. Next, we identified the challenges encountered in the implementation of a block-based version of the MC-SURE algorithm. We then proposed a framework that addresses these challenges and described how we can select various parameters used in our approach. Finally, we demonstrated that our approach can be used successfully to denoise a real dataset that fits the data model, where the noise is additive white Gaussian noise.

ACKNOWLEDGMENT

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

REFERENCES

- [1] A. Rizzo and M. G. Xibilia, *An Innovative Intelligent System for Sensor Validation in Tokamak Machines.*, IEEE Transactions of Control Systems Technology, Vol. 10, No. 3, May 2002.
- [2] U. Sulayman and A. T. Alouani, *Smart Grid Monitoring using Local Sensor Network: Real-time Data Acquisition, Analysis and Management.*, 2011 Proceedings of IEEE Southeastcon, Pages 444-449, March 2011
- [3] K. Kwong, R. Kavalier, R. Rajagopal and P. Varaiya, *Real-time Measurement of Link Vehicle Count and Travel Time in Road Network.*, IEEE Transactions of Intelligent Transportation Systems, Vol. 11, No. 4, Pages 814-825, Dec. 2011.
- [4] M. Ndoye, A. M. Barker, J. V. Krogmeier and D. M. Bullock, *A Recursive Multiscale Correlation-Averaging Algorithm for an Automated Distributed Road-Condition-Monitoring System.*, IEEE Transactions of Intelligent Transportation Systems, Vol. 12, No. 3, Pages 795-808, Sep. 2011.
- [5] S. Kay, *Fundamentals of Statistical Signal Processing, Volume 1: Estimation Theory*, Prentice Hall 1993.
- [6] S. Ramani, T. Blu and M. Unser, *Monte-Carlo SURE: A Black-Box Optimization of Regularization Parameters for General Denoising Algorithms*, IEEE Transactions of Image Processing, Vol. 17, No. 19, September 2008.
- [7] C. Stein, *Estimation of the Mean of a Multivariate Normal Distribution*, Annals of Statistics, Vol. 9, Pages 1135-1151, 1981.
- [8] D. L. Donoho, *Adapting to Unknown Smoothness via Wavelet Shrinkage*, J. Amer. Statist. Assoc., Vol. 90, No. 432, Pages 1200-1224, 1995.
- [9] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms and Applications. 4th Edition*, Prentice Hall, 2007.
- [10] S. Orfanidis, *Introduction to Signal Processing*, Prentice Hall, 1995.
- [11] A. P. Witkin, *Scale-Space Filtering*, Proc. 8th Int. Joint Art. Intell., Karlsruhe, Germany, Pages 1019-1022, 1983.
- [12] T. Lindeberg, *Scale Space Theory: A Basic Tool for Analyzing Structures at Different Scales*, Journal of Applied Statistics, Vol. 21, No. 2, Pages 224-270.